

Mateusz Stasielowicz
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
mateusz.stawicz@gmail.com

Kompresja obrazów z wykorzystaniem kompresji fraktalnej i systemu funkcji iterowanych

1. Wprowadzenie

Przetwarzanie i przechowywanie informacji towarzyszyło człowiekowi od zawsze w każdej dziedzinie życia. Obecnie większość danych zapisana jest na nośnikach cyfrowych jako ciągi bitów a ilość przechowywanych informacji jest ogromna i rośnie w bardzo szybkim tempie. W każdym obszarze działalności ludzkiej gdzie istnieje potrzeba magazynowania i przesyłania informacji znajdują zastosowanie algorytmy kompresji danych. Dzięki efektywnemu wykorzystaniu, algorytmy te są w stanie zmniejszyć ilość przechowywanych danych kilkukrotnie co przekłada się na duże oszczędności związane ze składowaniem i przesyłaniem danych. Obecnie kompresja obrazów zdominowana jest przez algorytm JPEG, alternatywę do niego może stanowić kompresja fraktalna, która szersze zastosowanie może znaleźć właśnie w kompresji obrazów. Metoda ta może dawać zaskakująco dobre wyniki w kompresji obrazu zachowując jednocześnie dobrą jakość skompresowanego obrazu.

2. Zastosowania kompresji fraktalnej

Istnieją dwie główne kategorie podziału metod kompresji – stratne i bezstratne. Wynikiem działań metod bezstratnych jest obraz który po dekompresji będzie identyczny jak obraz oryginalny. Stratne metody kompresji różnią się pod tym względem. Obraz skompresowany nie jest identyczny z oryginałem, jest tylko w pewnym stopniu podobny. Największą przewagą algorytmów stratnych nad bezstratnymi jest poziom kompresji który jest kilkukrotnie bądź kilkunastokrotnie większy na korzyść algorytmów stratnych. Kompresja fraktalna zaliczana jako metoda kompresji stratnej może być zastosowana wszędzie tam gdzie dopuszczalna jest utrata części informacji podczas procesu kompresji. Z tego względu wykorzystanie metod fraktalnych jest możliwe dla tych informacji, które mogą zostać zniekształcone, a ludzkie zmysły ze względu na swoje niedoskonałości nie są w stanie wychwycić różnic pomiędzy zniekształconą informacją skompresowaną

a oryginalną. Algorytmy kompresji stratnej znajdują w większości zastosowanie w kompresji obrazu, filmu i dźwięku.

Kompresja fraktalna bazuje na pojęciu fraktala. Fraktale zostały wprowadzone do matematyki w latach 70. XX wieku przez Benoîta Mandelbrota i do dzisiaj są przedmiotem wielu badań na całym świecie. Kompresja fraktalna wykorzystuje jedną z głównych cech fraktali – samopodobieństwo – jest to własność zbioru polegająca na tym, że dowolny mały fragment tego zbioru jest podobny w danej skali do większego fragmentu samego siebie. Bazując na powyższym stwierdzeniu można dojść do wniosku, że kompresja fraktalna może być wykorzystywana wyłącznie tam gdzie w kompresowanym obiekcie występują elementy samopodobne. W większości obrazów i filmów występuje bardzo duża ilość powtórzeń afinicznych, dzięki czemu można zastosować względem nich metody kompresji fraktalnej.

3. Algorytm kompresji fraktalnej

Algorytm wykorzystany w przeprowadzonych badaniach bazuje na podstawowych założeniach kompresji fraktalnej. Do wyszukiwania elementów podobnych w obrazie służą przekształcenia afiniczne, składają się one z przekształceń liniowych i translacji.

$$w = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (1)$$

$$w(x, y) = (ax + by + e, cx + dy + f) \quad (2)$$

Przy czym:

- w jest przekształceniem afinicznym,
- a, b, c, d odpowiadają za przekształcenia liniowe takie jak skalowanie i obrót,
- e, f odpowiadają za translację odpowiednio punktów x i y

3.1. Założenia wstępne

Przed przeprowadzeniem badań zostały zdefiniowane ograniczenia i wartości początkowe niektórych parametrów. Zdecydowano się na zastosowanie obrazu w niskiej rozdzielczości i skali szarości ze względu na dużą złożoność obliczeniową algorytmu. Wykorzystany obraz wejściowy jest przedstawiony na rysunku poniżej (Rys. 1). Założenia wstępne:

- Obraz kompresowany jest w skali szarości,
- Obraz kompresowany jest w rozdzielczości 256x256 co daje w sumie 65536 pikseli,
- Obraz wymaga 65 536 bajty pamięci,

- Współczynnik skalowania jest równy 0,5 dla wszystkich przekształceń afinicznych,
- Zostały zdefiniowane cztery przekształcenia obrotu (0° , 90° , 180° , 270°),
- Obrazem kompresowanym jest obraz "lenna",
- Bloki i obszary są o kształcie kwadratu.



Rys. 1. Obraz "Lenna" z przykładowymi elementami samopodobnymi

3.2. Opis algorytmu

Pierwszym krokiem działania algorytmu jest podział obrazu wejściowego na nienachodzące na siebie bloki. Bloki te mogą być dowolnego kształtu i rozmiaru. W algorytmie zostały przyjęte bloki o rozmiarze 8×8 pikseli w ten sposób zostały zdefiniowane 1024 bloki. Następny krok to zdefiniowanie obszarów o większym rozmiarze niż bloki. Dla uproszczenia współczynnik skalowania został zdefiniowany jako 0,5 dzięki czemu wszystkie obszary mają rozmiar 16×16 pikseli. Obszary mogą zachodzić na siebie, nie jest wymagane również aby pokrywały one cały obraz. W przypadku definiowania obszarów z krokiem co 1 piksel zostanie zdefiniowanych 58 081 obszarów. Zmniejszenie ilości obszarów skutkowało by pogorszeniem jakości skompresowanego obrazu lecz znacznie przyspieszyłoby czas działania algorytmu.

W kolejnym kroku algorytm porównuje wszystkie bloki z każdym z obszarów w każdej z 4 sekwencji obrotu. Jest to najbardziej czasochłonny etap działania algorytmu ze względu na dużą ilość niezbędnych do wykonania porównań. W przypadku obrazu w skali szarości, przekształcenie afiniczne można wyrazić w postaci:

$$W = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix} \quad (3)$$

Przy czym:

- w jest przekształceniem afinicznym,
- a, b, c, d odpowiadają za przekształcenia liniowe takie jak skalowanie i obrót,
- e, f odpowiadają za translację odpowiednio punktów x i y.
 $e, f \in [-241, 241]$, $e, f \in Z$
- z jest poziomem szarości, gdzie $z \in [0, 255]$, $z \in Z$
- $s \in [0, 1]$, $s \in R$
- $o \in [-255, 255]$, $o \in Z$

Porównanie bloku R i obszaru D odbywa się poprzez wyznaczenie współczynników s i o dla każdego z wariantów obrotu obszaru D, a następnie obliczenia średniego błędu kwadratowego g_{rms} . Znalazienie najlepiej pasującego obszaru D do bloku R polega na minimalizacji g_{rms} .

$$g_{rms} = \sqrt{\sum_{k=1}^{N^2} (s * d_k + o - r_k)^2} \quad (4)$$

$$s = \frac{N^2 \sum_{k=1}^{N^2} d_k r_k - \sum_{k=1}^{N^2} d_k \sum_{k=1}^{N^2} r_k}{N^2 \sum_{k=1}^{N^2} d_k^2 - \left(\sum_{k=1}^{N^2} d_k\right)^2} \quad (5)$$

$$o = \frac{\sum_{k=1}^{N^2} r_k - s \sum_{k=1}^{N^2} d_k}{N^2} \quad (6)$$

Przy czym:

- d_k jest poziomem szarości k-tego punktu w bloku D
- r_k jest poziomem szarości k-tego punktu w obszarze R
- N jest długością boku bloku R

Działanie algorytmu kończy się wraz z wyznaczeniem całego wektora przekształceń afinicznych W. W przypadku tego wariantu algorytmu wektor W będzie zawierał 1024 przekształcenia afiniczne.

3.3. Wynik działania algorytmu

Cała skompresowana informacja zawiera się w wektorze przekształceń W . Na tym etapie możliwe jest obliczenie ilości bitów niezbędnych do zakodowania niniejszego wektora. Wszystkie niezbędne informacje do przeprowadzenia dekompresji zawierają się w przekształceniach afinicznych i współczynnikach s i o co zostało przedstawione w poniższej tabeli (Tab. 1).

Tab. 1. Zbiór danych przedstawiający ilość bitów potrzebnych do zakodowania jednego bloku

Operacja	Ilość bitów potrzebnych do zapisu
Translacja x	9 bitów
Translacja y	9 bitów
Skalowanie (stała wartość)	0 bitów
Rotacja (4 warianty obrotu)	2 bity
Współczynnik s	5 bitów
Współczynnik o	9 bitów
Suma	34 bity

W przypadku tego wariantu algorytmu zostały zdefiniowane 1024 bloki R . Dany obraz wejściowy który zajmował początkowo 65535 bajty, po skompresowaniu będzie reprezentowany przez 34816 bitów czyli 4351 bajty. Dzięki tym informacjom w łatwy sposób można obliczyć stopień kompresji, który w tym przypadku wynosi 15. Na rysunkach (Rys. 2 i 3) zostały przedstawione obraz przed kompresją i po kompresji. Można zauważyć, że obraz oryginalny jest lepszej jakości niż skompresowany, jednakże jest to widoczne dopiero po dokładniejszej analizie obu obrazów. Działanie algorytmu było testowane na jednostce komputerowej z procesorem Intel i7 o taktowaniu 2,7GHz, gdzie czas działania tej wersji algorytmu wyniósł około 263 sekundy. Szczegółowe informacje na temat złożoności obliczeniowej oraz czasu działania w zależności od rozdzielczości obrazu wejściowego zostały przedstawione w poniższej tabeli (Tab. 2).



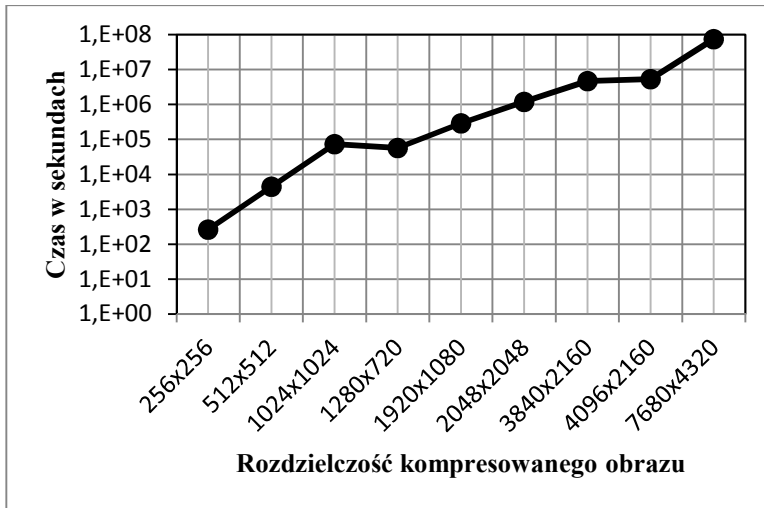
Rys. 2. Obraz wejściowy (65536 bajty)



Rys. 3. Obraz skompresowany (4351 bajty)

Tab. 2. Zbiór danych przedstawiający ilość porównań i szacunkowy czas kompresji dla wybranych rozdzielczości obrazu w najbardziej czasochłonnym etapie algorytmu z założonymi wcześniej parametrami

Rozdzielczość obrazu	Ilość bloków	Ilość obszarów	Ilość porównań	Szacunkowy czas działania w sekundach	Szacunkowy czas działania
256x256	1024	58081	2,37900E+08	263	4,4 minuty
512x512	4096	247009	4,04700E+09	4474	74,5 minuty
1024x1024	16384	1018081	6,67210E+10	73761	20,5 godziny
1280x720	14400	891825	5,13691E+10	56789	15,8 godziny
1920x1080	32400	2028825	2,62936E+11	290677	3,36 dnia
2048x2048	65536	4133089	1,08346E+12	1197778	13,86 dnia
3840x2160	129600	8204625	4,25328E+12	4702031	54,42 dnia
4096x2160	138240	8753745	4,84047E+12	5351177	61,93 dnia
7680x4320	518400	32997825	6,84243E+13	75643570	875,50 dnia



Rys. 4. Wykres zależności czasu kompresji od rozdzielczości kompresowanego obrazu, w skali logarymicznej

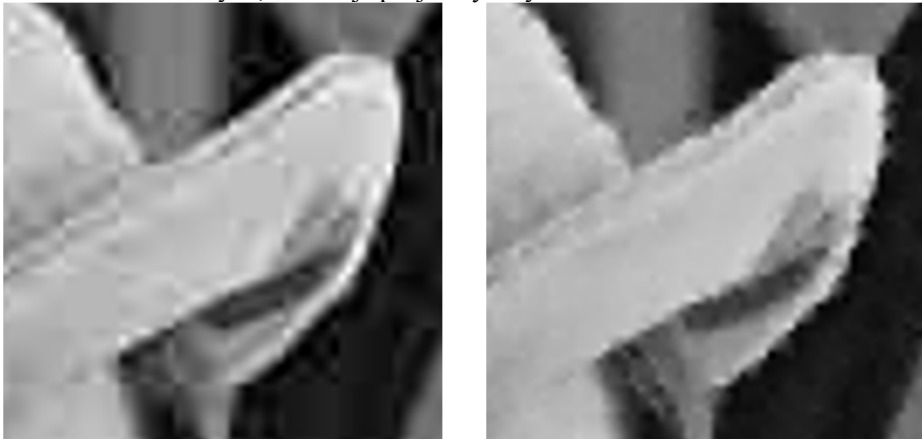
4. Kompresja fraktalna a JPEG

Najpopularniejszym obecnie sposobem kompresji obrazów jest algorytm JPEG. Bazuje on na dyskretnej transformacji kosinusowej, która jest stosowana między innymi w kompresji obrazów i filmów. JPEG charakteryzuje się dużą szybkością zarówno procesu kompresji jak i dekompresji. Jakość kompresji jest bardzo dobra przy niskim stopniu kompresji. Przy wyższym stopniu kompresji mogą pojawiać się tzw. artefakty, w obrazie mogą być widoczne poszczególne bloki pikseli o wymiarach 8x8 co powoduje znaczne pogorszenie jakości lecz skutkuje znacznie zmniejszoną ilością pamięci potrzebną do zapisu obrazu. JPEG wykazuje gorszą jakość obrazu skompresowanego przy kompresji obrazu na którym widnieje tekst, lub obrazów z nagłymi zmianami kontrastu i wyraźnymi podziałami na obrazie.

Kompresja fraktalna zalicza się do asymetrycznych algorytmów kompresji, oznacza to iż czas procesu kompresji różni się w znacznym stopniu od czasu dekompresji. W przypadku kompresji fraktalnej, czas procesu kompresji nieporównywalnie większy od czasu dekompresji, spowodowane jest to złożonością obliczeniową i ilością porównań które trzeba przeprowadzać pomiędzy blokami i obszarami obrazu. Czas jest główną wadą kompresji fraktalnej. Jednakże posiada ona wiele zalet. Jakość skompresowanego obrazu w zależności od parametrów algorytmu może być bardzo dobra przy dużym stopniu kompresji, lecz przekłada się to na dłuższy czas działania algorytmu. W odróżnieniu od JPEG, nie mają tutaj znaczenia duże, nagłe zmiany kontrastu w obrazie, ponadto kompresja fraktalna jest niezależna od rozdzielczości – jest to jedna z cech obiektów fraktalnych, oznacza to

iz obraz może być dekompresowany do dowolnej wielkości bez strat na jakości obrazu, czego nie można zrobić za pomocą algorytmu JPEG. Na poniższym rysunku (Rys. 5.) można zaobserwować różnice pomiędzy kompresją fraktalna i JPEG. Aby różnice były lepiej uwidocznione, obraz został skompresowany algorytmem JPEG z parametrem jakości 20 tak aby ilość bajtów niezbędna do zapisu była taka sama jak obrazu skompresowanego metodami fraktalnymi. Zostały również powiększone odpowiadające sobie elementy obu obrazów.

Algorytm JPEG jest obecnie najpopularniejszym standardem kompresji stratnej obrazów na świecie. Jest obsługiwany przez wiele urządzeń, stron internetowych, aplikacji. W wielu miejscach stał się standardowym formatem za pomocą którego reprezentowane są obrazy, jak choćby w telefonach komórkowych, wielu aparatach fotograficznych i stronach internetowych. Wszystko to sprawia, że bardzo trudno byłoby zastąpić to rozwiązanie innym, jednakże kompresja fraktalna może znaleźć zastosowanie w innych, bardziej specjalistycznych obszarach.



Rys. 5. Fragment obrazu "Lenna" przybliżenie 4 krotne. Obraz po lewej skompresowany za pomocą JPEG o jakości 20 (4417 bajtów). Obraz po prawej skompresowany kompresją fraktalną (4351 bajtów)

5. Podsumowanie

W pracy zaprezentowano metodę stratnej kompresji obrazów z wykorzystaniem metod fraktalnych. Przedstawiono również ogólny sposób działania algorytmu kompresji fraktalnej oraz bardziej obfityjający w szczegóły opis badanego wariantu algorytmu wraz z wynikami jego działania.

Kompresja obrazów metodami fraktalnymi daje bardzo dobre rezultaty zarówno w jakości kompresowanego obrazu jak i stopniu kompresji. Porównanie z algorytmem JPEG pokazało iż fraktalne metody kompresji obrazów wolne są od

wad jakie niesie ze sobą kompresja JPEG i wykorzystana w niej dyskretna transformata kosinusowa. Ponadto metody fraktalne dają dodatkowo możliwość dekompresji obrazu do dowolnej wielkości bez utraty jakości co jest cechą wszystkich obiektów fraktalnych. Mimo swoich zalet metody kompresji fraktalnej posiadają jedną bardzo znaczącą wadę, co zostało wykazane w badaniu. Metody kompresji fraktalnej posiadają bardzo dużą złożoność obliczeniową potrzebną do wykonania procesu kompresji obrazu co przekłada się bezpośrednio na bardzo długi czas działania algorytmu. Czas kompresji wyklucza obecnie algorytmy fraktalne z zastosowania tam gdzie kompresja obrazu powinna odbywać się w czasie rzeczywistym.

Aby poprawić wydajność algorytmu kompresji fraktalnej możliwe jest przeprowadzenie jego optymalizacji. Algorytm posiada wiele parametrów takich jak wielkość bloków i obszarów oraz ich kształt, odległości między blokami, warianty obrotu obszarów, współczynnik skalowania obszaru do bloku. Parametry te mogą zostać dobrane w sposób optymalny tak aby jak najbardziej skrócić czas działania algorytmu bądź też poprawić jakość skompresowanego obrazu, jednakże optymalizacja wieloparametrowa jest niezwykle trudnym zagadnieniem. Innym sposobem optymalizacji jest wstępna analiza obrazu i jego segmentacja na obszary podobne, które będą kompresowane niezależnie od siebie, równoległe z pozostałymi obszarami. Kolejnym sposobem optymalizacji jest odpowiednie zawężenie obszaru poszukiwań bloku podobnego tak aby jak najbardziej zmniejszyć złożoność obliczeniową algorytmu pozostawiając ten sam stopień kompresji obrazu. Możliwe jest również zastosowanie algorytmów genetycznych, które dobrze sprawdzają się przy wyszukiwaniu rozwiązań optymalnych w dużych zbiorach rozwiązań. Dodatkowo algorytmy genetyczne mogą być w łatwy sposób zrównoleglone co może mieć znaczący wpływ na szybkość działania algorytmu. Dalszy rozwój algorytmu kompresji fraktalnej związany jest głównie z badaniami nad metodami jego optymalizacji, w szczególności nad zastosowaniem algorytmów genetycznych oraz wstępnej analizy obrazu, które znacząco mogą wpłynąć zarówno na szybkość działania algorytmu jak i polepszyć stopień kompresji.

Literatura

1. Yuval Fisher: *Fractal Image Compression Theory and Application*. Springer-Verlag 1995
2. Stephen Welstead: *Fractal and Wavelet Image Compression Techniques*. SPIE Washington 1999
3. William B. Pennebaker, Joan L. Mitchell: *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers 2004

Streszczenie

Pod pojęciem kompresji obrazów kryją się najróżniejsze algorytmy kompresji danych. Najczęściej są to algorytmy kompresji stratnej, które charakteryzują się znacznym stopniem kompresji, lecz ich wadą jest utrata informacji podczas procesu kompresji. W niniejszym artykule zostanie opisany algorytm kompresji fraktalnej, który zalicza się do algorytmów kompresji stratnej i najczęściej wykorzystywany jest przy kompresji obrazów. Zostanie również przedstawiony opis działania algorytmu z wykorzystaniem systemu funkcji iterowanych oraz jego wady i zalety. Celem artykułu jest porównanie możliwości kompresji metodą fraktalną z najpopularniejszą obecnie metodą kompresji stratnej obrazów JPEG. Uzyskane dane posłużą do dalszej analizy możliwości algorytmu kompresji fraktalnej oraz metod jego optymalizacji.

Słowa kluczowe: kompresja fraktalna, kompresja obrazów, kompresja stratna, fraktal

Summary

The term image compression hides various data compression algorithms. Usually these are lossy compression algorithms, which characterize high degree of compression, but their disadvantage is the loss of information during compression process. The article describes fractal compression algorithm, which is one of the lossy compression algorithms and is used mostly in image compression. It will be also described algorithm details using iterated function system and its advantages and disadvantages. The aim of the article is to compare the capabilities of fractal compression method with JPEG which is currently the most popular method of lossy image compression. The obtained data will be used to further analyze possibilities of fractal compression and methods of its optimization.

Keywords: fractal compression, image compression, lossy compression, fractal